



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER OF PATENTS AND TRADEMARKS  
Washington, D.C. 20531  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/204,973	12/03/1998	DAVID LARS EHNEBUSKE	AT9-98-266	9309

7590 05/07/2002

DUKE W YEE  
CARSTENS YEE AND CAHOON  
P O BOX 802334  
DALLAS, TX 75380

EXAMINER

INGBERG, TODD D

ART UNIT PAPER NUMBER

2124

DATE MAILED: 05/07/2002

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

Application No.  
09/204,973

Applicant(s)  
David Lars Ehnebuske et al.

Examiner  
Todd Ingberg

Art Unit  
2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136 (a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on Feb 14, 2002.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11; 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-28, 31-39, 41-52, 55-63, 65-76, 79-87, 89-98 is/are pending in the application.
- 4a) Of the above, claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☒ Claim(s) 2-11, 48-50, 70, and 94 is/are allowed.
- 6) ☒ Claim(s) 1, 12 - 28, 31-39, 41-47, 51-52, 55-63, 65-69, 71-76, 79-87, 89-93, 9 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claims \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are objected to by the Examiner.
- 11) ☐ The proposed drawing correction filed on \_\_\_\_\_ is: a) ☐ approved b) ☐ disapproved.
- 12) ☐ The oath or declaration is objected to by the Examiner.

## Priority under 35 U.S.C. § 119

- 13) ☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).
- a) ☐ All b) ☐ Some\* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \*See the attached detailed Office action for a list of the certified copies not received.
- 14) ☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

## Attachment(s)

- 15) ☐ Notice of References Cited (PTO-892)
- 16) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 17) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s). \_\_\_\_\_
- 18) ☐ Interview Summary (PTO-413) Paper No(s). \_\_\_\_\_
- 19) ☐ Notice of Informal Patent Application (PTO-152)
- 20) ☐ Other:

Art Unit: 2124

### **DETAILED ACTION**

Claims 29, 30, 40, 53, 54, 64, 77, 78 and 88 have been canceled.

Claims 1-2, 12, 23, 27, 36, 46, 48, 51, 60, 70, 72, 75, 84, 94 and 96 have been amended.

Claims 2 - 11, 48-50, 70 and 94 are allowed.

Claims 1 - 28, 31- 39, 41-52, 55-63, 65-76, 79-87, 89- 98 have been examined.

### ***Information Disclosure Statement***

1. Examiner has made of record documents on the products listed in prior office action and made Applicant aware of the products. Based on the Applicant's response the products are not related.

### ***Specification***

2. The substitute paragraph has been entered and has corrected the prior error in the Specification.

### ***Examiner's Interpretation***

3. The following are Examiner interpretations.

a. ***Modeling*** - The Martin book teaches the use of modeling an enterprise ( **Martin**, page 247 - 249) operation and provides Appendix A, Recommended Diagramming Standards. Martin page 285 illustrates the transition from the problem domain, to modeling, to OO Design to code. The Martin reference has many chapters covering modeling and discusses the models are tied together to generate code (**Martin**, page 155, box).

Art Unit: 2124

The Martin reference also provides some examples. These models are not viewed as static. The Martin reference teaches modeling the enterprise. The principles and techniques are dynamic.

b. **Flow Control** - The Applicant does not explicitly claim flow control. However, when the Applicant states in the claims “method logic is continuous” this is interpreted as meaning the method (a common feature in object technology) can run until a outside interrupt occurs. This is a product of flow control resulting from the logic structure of a computer program. There are many claims to what the Examiner interprets as claims to flow control. Flow control is the tracing the path of an executing of a program. The exact path the execution of a program will follow is determined by the values of the attributes and the control conditions encountered. The examples of the programming constructs such as Martin page 148 show the difference paths flow control can take depending on execution of operations such as CHECK IN COPY , FILL REQUEST and BOOK OVERDUE. The values of attributes are tested to determine the path taken. Many claims have made claim to flow control which is inherent to the execution of a computer programs.

c. Claim 3 contains the following limitation “.... the step of defining a first control point further comprises:

(a1) decorating the object to dynamically insert a first control point such that the object acquires this new control point.”

The Examiner interprets the “decorating” to mean the entry of programming information such as the operations/methods and the entry of control points in a programming environment.

Art Unit: 2124

*Claim Rejections - 35 USC § 102*

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. § 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1 - 98 are rejected under 35 U.S.C. 102(a) as being anticipated by “Principles of Object-Oriented Analysis and Design”, James **Martin**, published June 1, 1992.

The Martin teaches the underlying theory of building an Object Oriented Computer Aided Software Engineering (OO-CASE) tools in his 1992 text book. The Martin should be taken as a whole, however, focus of the rejection is on the Chapters 9 and 10. The Martin references covers the very basics of object technology that one of ordinary skill should have known well before the time of invention:

Chapter 2 - Basic Concepts

Chapter 3 - Why Object-Oriented ?

Chapter 4 - Basic Guidelines

Chapter 6 - Categorizing Objects

Chapter 7 - Relationships Among Object Types

Chapter 8 - State and State Changes

**Chapter 9 - Events, Triggers, and Operations**

Art Unit: 2124

**Chapter 10 - Rules**

Chapter 11 - How Diagrams Interrelate

Chapter 12 - Basic Concepts of OO Design

Chapter 15 - Method Creation

Chapter 18 - OO-CASE Tools

Appendix A - Recommended Diagramming Standards

The Martin book in addition to containing foundation knowledge of object oriented technology it teaches applying a set of rules comprising the placing of logic (program statements) in a pre-method control before the logic of a method and post method control point after the logic of a method. Martin also teaches associating a set of rules with each control point based on the class of the object in which the method resides, name of the method and type of control point and invoking methods.

**Claim 1**

**Martin** anticipates a computer implemented process for applying a set of rules (**Martin**, Chapter 10, RULES, and page 138-139 and 249-251), the process comprising:

- (a) placing a pre-method control point before logic of a method (**Martin**, page 142, operation precondition) and post-method control point after the logic of the method (**Martin**, page 142, postcondition )
- (b) associating a set of rules with each control point (**Martin**, page 142, 147 "Operation" as per (a) above) based on a class of object in which the method resides (**Martin**, page 143, "... rules

Art Unit: 2124

associated with diagrams of OO ...” ) , name of the method, and type of control point, whether the pre-method control point or the post-method control point (**Martin**, page 142, operation precondition) ;

(c) invoking the method ( **Martin**, page 116), wherein encountering each control point during the execution of the method comprises (**Martin**, page 142, postcondition ):

(i) determining if the encountered control point is active (**Martin**, page 122, IF structure in center diagram ) ;

(ii) on the basis of an active control point (Interpreted as the result of the IF structure above further described in Appendix A on page 381 Control Conditions):

1) selecting rules based on a set of rules associated with the active control point associated in step (**Martin**, page 122, first diagram example is the control condition to fire missile ) (b);

2) running the selected rules (**Martin**, page 122, rule that lead to the control condition );

3) obtaining results from running the rules (**Martin**, page 122, trigger rule at the bottom of the page ); and

4) combining the results using a combining algorithm specified by the control point (**Martin**, page 122, A control condition can function as a combining algorithm as seen in diagram in middle of the page and page 126 Figure 9.9 and **Martin** teaches a way to have a combining algorithm where one of three operations are selected as on page 124, and **Martin** teaches a way to have a combining algorithm where one can be selected as taught in the mutually exclusive notation on the bottom of page 125).

Art Unit: 2124

**Claim 12**

**Martin** anticipates a computer implemented process for applying a set of rules ( as per claim 2), comprising:

- (a) invoking a method in an object ( as per claim 2);
- (b) encountering an active control point during the invocation of the method, wherein the method is a context control point ( as per claim 2);
- (c) selecting rules associated with the method of the object at the control point ( as per claim 2);
- (d) invoking the rules ( as per claim 2); and
- (e) combining results from invoking the rules as per claim 1.

**Claim 13**

The process of claim 12, wherein the rules perform a variety of actions (Martin, page 164, a variety of actions can occur such as Invoice Student OR Get Dorm depending on the outcome of the Remote Student registered condition ) conditioned by the fact that rules may be associated with particular, regularly occurring points in the object model Martin,

(**Martin**, page 166, RULES LINKED TO DIAGRAMS, “The importance of rules was emphasized in Chapter 10 which indicated that rules can be connected to any of the OO diagrams”.)

**Claim 14**



Art Unit: 2124

The process of claim 12, wherein the rules perform at least one function which varies over time (**Martin**, page 117, clock events and page 144 Rules Associated with Event Diagrams - “ If time is between 9 AM to 5 PM”, **Martin**, page 394, Clock Events).

**Claim 15**

A process of claim 12, wherein a control point occurs just before logic of the method begins, just after the logic of the method completes, or at both just before logic of the method begins and just after the logic of the method completes as per claim 1.

**Claim 16**

**Martin** anticipates a computer implemented process for applying a set of rules ( as per claim 2) comprising:

- (a) defining an object ( as per claim 2);
- (b) defining at least one method in the object ( as per claim 2);
- (c) defining at least one control point in the at least one method ( as per claim 2).
- (d) defining rules to the at least one control point on basis the object's class name, method, name, and position of the at least one control point in the method (**Martin**, Chapter 12, BASIC CONCEPTS OF OO DESIGN, page 172 - 173, the relationship between classes and objects and the relationship between rules and object modeling ).

**Claim 17**

In the process of claim 16, further comprising the step of activating at least one control point having associated rules as per claim 1.

Art Unit: 2124

**Claim 18**

The process of claim 16 further comprises:

- (e) encountering a first control point (**Martin**, page 173, control point with a Time Event) ;
- (f) running the rules associated with the first control point (**Martin**, page 173, control point with a Time Event); and
- (g) affecting behavior of the object based on running the rules associated with the first control point (The flow control is controlled by the Rule associated with the Control point as per **Martin**, page 381).

**Claim 19**

In the process of claim 18, the step of affecting the behavior of the object further comprises:

- (h) associating different rules to a control point ( as per claim 14 - Different rules based on the time of day affects the flow control/ behavior).

**Claim 20**

In the process of claim 18, the step of affecting the behavior of the object further comprises:

- (h) defining another control point (Examiner Interpretation of “defining another control point” the meaning could be at design time or runtime. Design time would involve the interaction with the OO-CASE tool as on **Martin**, page 162, Run time would mean the behavior changes value such as attribute which influence the path the control flow takes. This is the point of programming. The ability to model a problem domain and execute code that process information that reflects the

Art Unit: 2124

modeled problem - Flow Control as determined by the running of the program such as **Martin**, page 163).

**Claim 21**

In the process of claim 18, the step of modifying the object further comprises:

(h) associating rules to a second control point ( **Martin**, page 163, Multiple control points defined).

**Claim 22**

In the process of claim 16, further comprising a step of deactivating the at least one control point.( As per claim 1. The control point is determined if it is active or not. If one takes the **Martin**, page 163 example where the timed event is part of the Waitlisted functionality the Timed event occurs at a specific time the Timed Event is one example of activating and deactivating the control point also see **Martin**, Appendix A, page 394)

**Claim 23**

**Martin** anticipates a computer implemented process for applying a set off rules ( as per claim 2), comprising

- (a) defining an object ( as per claim 2);
- (b) defining a method in the object ( as per claim 2);
- (c) defining a first control point of the method, the first control pint being a method context control point ( as per claim 2);
- (d) determining rules associated with the first control point ( as per claim 2);

Art Unit: 2124

(e) defining a second control point of the method the second control pint being a method context control point( as per claim 2); and

(f) determining rules associated with the second control point ( as per claim 2).

**Claim 24**

A computer implemented process as in claim 23 further comprising:

(g) separately selecting, running and combining the results of rules determined to be associated with either control point as per claim 1.

**Claim 25**

In the process of claim 23 wherein the first control point is a pre-method trigger point ( **Martin**, page 142, diagram top of page, page 381 Trigger Rules).

**Claim 26**

In the process of claim 23 wherein the second control point is a post-method trigger point (**Martin**, page 115, Postconditions in cause and effect isolation, page 141, Post Condition page 381, Trigger Rule).

**Claim 27**

**Martin** anticipates a computer implemented process for defining an object (Martin, page 166 - 167, Link between, Diagrams, Rules and Objects ) comprising:

defining an object; (**Martin**, page 144, Box 10.3, and page 169 - 176 and as per claim 2)

defining a method in the object by: defining method logic ( as per claim 2) ;

Art Unit: 2124

placing the method logic in the method (**Martin**, page 173, methods is the Specification of an operation and as per claim 2);

defining at least one control point wherein the at least one control point is a method context control point( as per claim 2);

and placing the at least one control point in the method wherein the method logic is continuous wherein the step of placing the at least one control point further comprises placing the at least one control in the method after the method logic. (**Martin**, page 224, DO and FOR loops, page 225, Loops in action diagrams).

**Claim 28**

A computer implemented process for defining an object as in claim 27, wherein the step of placing the at least one control point further comprises placing the at least one control in the method before the method logic ( as per claim 1).

**Claim 31**

A computer implemented process for defining an object as in claim 27, further comprises: flagging the at least one control point on the basis of being active ( as per claim 1).

**Claim 32**

A computer implemented process for defining an object as in claim 27, wherein the step of defining the at least one control point further comprising: defining a rule selection algorithm associated with the at least one control point (**Martin**, page 168, control point rule illustrated).

**Claim 33**

Art Unit: 2124

A computer implemented process for defining an object as in claim 27, wherein the step of defining the at least one control point further comprising: defining a rule result combination algorithm associated with the at least one control point. As per claim 1.

**Claim 34**

A computer implemented process for defining an object as in claim 27, wherein the step of defining the at least one control point further comprises: defining a rule selection algorithm for the at least one control point; and defining a rule result combination algorithm for the at least one control point As per claim 1.

**Claim 35**

A computer implemented process for defining an object as in claim 27, further comprising: associating at least one rule with the at least one control point. As per claim 32.

**Claim 36**

**Martin** anticipates a computer implemented process for defining a rule comprising: creating the rule (**Martin**, page 167, Rule Editor) ; associating the rule with an object class ( **Martin**, page 167, Figure 11.14 ); associating the rule with a method within the object class ( **Martin**, page 173, operations are methods); and associating the rule with an occurrence of a control point within the method and associating the rule with another method within the object class. ( **Martin**, page 168, Figure 11.16 ).

**Claim 37**

Art Unit: 2124

A computer implemented process for defining a rule as in claim 36 wherein the occurrence of the control point within the method being before method logic. As per claim 1.

**Claim 38**

A computer implemented process for defining a rule as in claim 36 wherein the occurrence of control point within the method being after method logic. As per claim 1.

**Claim 39**

A computer implemented process for defining a rule as in claim 36, further comprising: associating the rule with another object class ( **Martin**, page 267, the ability to access a method/Rule from more than one object and the concept of Reuse which is a key factor in object oriented technology **Martin**, page 248, Box 16.2 Maximize reusability) ( This claim could also be interpreted as claiming the principle of inheritance as described on **Martin**, page 266 - 268).

**Claim 41**

A computer implemented process for defining a rule as in claim 36, further comprising: associating the rule with another control point within the method of the object class (**Martin**, page 166 - 168, the rule associated to the control point , page 233, RULES)

**Claim 42**

**Martin** anticipates a computer implemented process for applying a set of rules ( as per claim 2), comprising: selecting an object class; selecting a method within the object class; invoking the method; processing rules associated with the method comprising: encountering a control point associated with the method; determining if the control point is active; and finding at least one rule

Art Unit: 2124

associated with an active control point. (Interpreted as the running of the code generated by claim 2).

**Claim 43**

A computer implemented process for applying a set of rules as in claim 42, wherein the step of finding at least one rule further comprises: accessing a selecting algorithm associated with the active control point ( as per claim 1); and selecting at least one rule using the selecting algorithm ( as per claim 10 and The IF structure in the control point as per **Martin**, page 168 ).

**Claim 44**

A computer implemented process for applying a set of rules as in claim 42, where in the step of processing rules further comprises: running the at least one rule; determining results from running the at least one rule; accessing a combining algorithm associated with the control point; and combining the results using the combining algorithm. As per claim 1.

**Claim 45**

**Martin** anticipates a computer implemented process for applying a set of rules, comprising: selecting an object class; selecting a method within the object class; invoking the method; processing rules comprising: encountering a control point; accessing a selecting algorithm associated with the control point; and selecting at least one rule using the selecting algorithm. As per claim 42.

**Claim 46**



Art Unit: 2124

**Martin** anticipates a computer implemented process for applying a set of rules, comprising: selecting an object class; selecting a method within the object class; invoking the method; processing rules comprising: encountering a method context control point; finding at least one rule associated with the method context control point; running the at least one rule; determining results on the basis of running the at least one rule; accessing a combining algorithm associated with the method context control point; and combining the results using the combining algorithm.

As per claim 1 - the running of the executable generated from the model.

**Claim 47**

**Martin** anticipates a computer implemented process for applying a set of rules, comprising: selecting an object class; selecting a method within the object class; invoking the method; processing rules comprising: encountering a first control point associated with the method; determining if the first control point is active (the running of code from claims 1 and 2 and implementations such as page 164 Fig 11.10); executing method logic of the method; encountering a second control point associated with the method; determining if the second control point is active; finding a set of rules associated with one of the first control point and the second control point, wherein the set of rules contains not less than zero rules as per claim 9.

**Claim 51**

**Martin** anticipates a data processing system for defining an object comprising: defining means for defining an object; defining means for defining a method in the object by: defining means for

Art Unit: 2124

defining method logic; placing means for placing the method logic in the method; defining means for defining at least one control point; and placing means for placing the at least one control point wherein the at least one control point is a method context control point; in the method wherein the method logic is continuous wherein the step of placing the at least one control point further comprises placing means for placing the at least one control in the method after the method logic.. As per claim 27.

**Claim 52**

A data processing system for defining an object as in claim 51, wherein the step of placing the at least one control point further comprises placing means for placing the at least one control in the method before the method logic. As per claim 1.

**Claim 55**

A data processing system for defining an object as in claim 51, further comprises: flagging means for flagging the at least one control point on the basis of being active. As per claim 31.

**Claim 56**

A data processing system for defining an object as in claim 51, wherein the step of defining the at least one control point further comprising: defining means for defining a rule selection algorithm associated with the at least one control point. As per claim 32.

**Claim 57**

A data processing system for defining an object as in claim 51, wherein the step of defining the at

Art Unit: 2124

least one control point further comprising: defining means for defining a rule result combination algorithm associated with the at least one control point as per claim 10.

**Claim 58**

A data processing system for defining an object as in claim 51, wherein the step of defining the at least one control point ( as per claims 1 and 2) further comprises: defining means for defining a rule selection algorithm for the at least one control point; and defining a rule result combination algorithm for the at least one control point. As per claim 34.

**Claim 59**

A data processing system for defining an object as in claim 51, further comprising: associating means for associating at least one rule with the at least one control point. As per claim 8.

**Claim 60**

**Martin** anticipates a data processing system for defining a rule comprising: creating means for creating the rule; associating means for associating the rule with an object class; associating means for associating the rule with a method within the object class; associating means for associating the rule with an occurrence of a control point within the method; and associating means for associating the rule with another method within the object class. As per claim 36.

**Claim 61**

A data processing system for defining a rule as in claim 60 wherein the occurrence of the control point within the method being before method logic. As per claim 1.

**Claim 62**

Art Unit: 2124

A data processing system for defining a rule as in claim 60 wherein the occurrence of control point within the method being after method logic. As per claim 1.

**Claim 63**

A data processing system for defining a rule as in claim 60, further comprising: associating means for associating the rule with another object class. ( **Martin**, page 267, the ability to access a method/Rule from more than one object and the concept of Reuse which is a key factor in object oriented technology **Martin**, page 248, Box 16.2 Maximize reusability) ( This claim could also be interpreted as claiming the principle of inheritance as described on **Martin**, page 266 - 268).

**Claim 65**

A data processing system for defining a rule as in claim 60, further comprising: associating means for associating the rule with another control point within the method of the object class. As per claim 1.

**Claim 66**

**Martin** anticipates a data processing system for applying a set of rules, comprising: selecting means for selecting an object class; selecting means for selecting a method within the object class; invoking means for invoking the method; processing means for processing rules associated with the method comprising: encountering means for encountering a control point associated with the method; determining means for determining if the control point is active; and finding means for finding at least one rule associated with an active control point. As per claim 42.

**Claim 67**

Art Unit: 2124

A data processing system for applying a set of rules as in claim 66, wherein the step of finding at least one rule further comprises: accessing means for accessing a selecting algorithm associated with the active control point; and selecting means for selecting at least one rule using the selecting algorithm. As per claim 43.

**Claim 68**

A data processing system for applying a set of rules as in claim 66, where in the step of processing rules further comprises: running means for running the at least one rule; determining means for determining results from running the at least one rule; accessing means for accessing a combining algorithm associated with the control point; and combining means for combining the results using the combining algorithm. As per claim 44.

**Claim 69**

**Martin** anticipates a data processing system for applying a set of rules, comprising: selecting means for selecting an object class; selecting means for selecting a method within the object class; invoking means for invoking the method; processing means for processing rules comprising: encountering means for encountering a control point; accessing means for accessing a selecting algorithm associated with the control point; and selecting means for selecting at least one rule using the selecting algorithm. As per claim 45.

**Claim 71**

**Martin** anticipates a data processing system for applying a set of rules, comprising: selecting means for selecting an object class; selecting means for selecting a method within the object class;

Art Unit: 2124

invoking means for invoking the method; processing means for processing rules comprising: encountering means for encountering a first control point associated with the method; determining means for determining if the first control point is active (as per claim 2); executing means for executing method logic of the method (as per claim 2); encountering means for encountering a second control point associated with the method; determining means for determining if the second control point is active; finding, means for finding a set of rules associated with one of the first control point and the second control point (as per claim 7), wherein the set of rules contains not less than zero rules. As per claim 9.

#### **Claim 72**

**Martin** anticipates a data processing system for applying a set of rules, comprising: selecting means for selecting an object class; selecting means for selecting a method within the object class; invoking means for invoking the method; processing means for processing rules comprising: encountering means for encountering a control point associated with the method; finding means for finding at least one rule associated with the control point prior to executing method logic of the method; running means for running the at least one rule; obtaining means for obtaining results on the basis of running the at least one rule; and controlling means for controlling the method on the basis of the results. As per claim 48.

#### **Claim 73**

A data processing system for applying a set of rules as in claim 72, wherein the step of controlling the method comprises: exiting means for exiting the method. As per claim 49.

Art Unit: 2124

**Claim 74**

A data processing system for applying a set of rules as in claim 72, wherein the step of controlling the method comprises: executing means for executing method logic of the method. As per claim 50.

**Claim 75**

**Martin** anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for defining an object, the instruction comprising: instructions for defining an object; instructions for defining a method in the object by: instructions for defining method logic; instructions for placing the method logic in the method; instructions for defining at least one control point; and instructions for placing the at least one control point in the method wherein the method logic is continuous. As per claim 51.

**Claim 76**

A computer program product for defining an object as in claim 75, wherein the instruction of placing the at least one control point further comprises placing the at least one control point in the method before the method logic. As per claim 1.

**Claim 79**

A computer program product for defining an object as in claim 75, further comprises: instructions for flagging the at least one control point on the basis of being active. As per claim 31.

**Claim 80**

Art Unit: 2124

A computer program product for defining an object as in claim 75, wherein the instruction of defining the at least one control point further comprising: instructions for defining a rule selection algorithm associated with the at least one control point. As per claim 32.

**Claim 81**

A computer product for defining an object as in claim 75, wherein the instruction of defining the at least one control point further comprises: instructions for defining a rule combination algorithm associated with the at least one control point. As per claim 33.

**Claim 82**

A computer program product for defining an object as in claim 75, wherein the step of defining the at least one control point further comprises: instructions for defining a rule selection algorithm for the at least one control point; and instructions for defining a rule result combination algorithm for the at least one control point. As per claim 34.

**Claim 83**

A computer program product for defining an object as in claim 75, further comprising: instructions for associating at least one rule with the at least one control point. As per claim 35.

**Claim 84**

**Martin** anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for defining a rule, the instruction comprising: instructions for creating the rule; instructions for associating the rule with an object class; instructions for associating the rule with a method within the object class; and instructions



Art Unit: 2124

for associating the rule with an occurrence of a control point within the method and instructions for associating the rule with another method within the object class.. As per claim 36.

**Claim 85**

A computer program product for defining a rule as in claim 84 wherein the occurrence of the control point within the method being before method logic. As per claim 1.

**Claim 86**

A computer program product for defining a rule as in claim 84 wherein the occurrence of control point within the method being after method logic. As per claim 1.

**Claim 87**

A computer program product for defining a rule as in claim 84, further comprising: instructions for associating the rule with another object class. As per claim 39 or 63.

**Claim 89**

A computer implemented process for defining a rule as in claim 84, further comprising: instructions for associating the rule with another control point within the method of the object class. As per claim 65.

**Claim 90**

**Martin** anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the instruction comprising: instructions for selecting an \object class; instructions for selecting a method within the object class; instructions for invoking the method; instructions for processing

Art Unit: 2124

rules associated with the method comprising: instructions for encountering a control point associated with the method; instructions for determining if the control point is active; and instructions for finding at least one rule associated with an active control point. As per claim 1.

**Claim 91**

A computer program product for applying a set of rules as in claim 90, wherein the step of finding at least one rule further comprises: instructions for accessing a selecting algorithm associated with the active control point; and instructions for selecting at least one rule using the selecting algorithm. As per claim 43.

**Claim 92**

A computer program product for applying a set of rules as in claim 90, where in the step of processing rules further comprises: instructions for running the at least one rule; instructions for determining results from running the at least one rule; instructions for accessing a combining algorithm associated with the control point; and instructions for combining the results using the combining algorithm. As per claim 1.

**Claim 93**

**Martin** anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the instruction comprising: instructions for selecting an object class; instructions for selecting a method within the object class; instructions for invoking the method; instructions for processing rules comprising: instructions for encountering a control point; instructions for accessing a

Art Unit: 2124

selecting algorithm associated with the control point; and instructions for selecting at least one rule using the selecting algorithm. As per claim 42.

**Claim 95**

**Martin** anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the instruction comprising: instructions for selecting an object class ( as per claim 42); instructions for selecting a method within the object class; instructions for invoking the method; instructions for processing rules comprising: instructions for encountering a first control point associated with the method; instructions for determining if the first control point is active; instructions for executing method logic of the method; instructions for encountering a second control point associated with the method; instructions for determining if the second control point is active (Asper claim 47); instructions for finding a set of rules associated with one of the first control point and the second control point, wherein the set of rules contains not less than zero rules. As per claim 9.

**Claim 96**

**Martin** anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the instruction comprising: instructions for selecting an object class; instructions for selecting a method within the object class; instructions for invoking the method; processing rules comprising: instructions for encountering a control point associated with the method, the control point being a method context control point; instructions for finding at least one rule associated with the control

Art Unit: 2124

point prior to executing method logic of the method; instructions for running the at least one rule; instructions for obtaining results on the basis of running the at least one rule; and instructions for controlling the method on the basis of the results. As per claim 48.

**Claim 97**

A computer program product for applying a set of rules as in claim 96, wherein the step of controlling the method comprises: instructions for exiting the method. **Martin**, page 236, use of “return” in C++ and it is well known in C++ that reaching the end of a method such as flow control reaching the last “}” in the method declassifies will return flow control to the method that called this method or terminate. In either path the method has performed an exit.

**Claim 98**

A computer program product for applying a set of rules as in claim 96, wherein the step of , controlling the method comprises: instructions for executing method logic of the method. As per claims 50.

***Allowable Subject Matter***

6. Claims 2 - 11, 48-50, 70 and 94 are allowed.

***Summary***

7. The rejection under Martin is a fundamental teaching from June 1992 which contains features inherent to Object Oriented technology and a primer book on the technology. The WFT rejection is a commercial product for use and for sale since April 1997 which implementations

Art Unit: 2124

Rules in an Object Oriented CASE tool. Both rejections support the concept that the tools are designed for the end user to implement rules without formal programming experience. Rational Rose is also a OO-CASE tool which teaches the ability implement rules at a Programmer level.

*Response to Arguments*

8. Applicant's arguments filed February, 14, 2002 have been considered and are partially persuasive.

**I. Applicant's Response to Examiner's Interpretation**

In this section of the Office Action, the Examiner sets forth the Examiner's "interpretations" of the terms "modeling" and "Flow Control." The Examiner's "interpretations" of these terms in no way limits Applicants' claimed invention. The terms in the claims must be interpreted in light of the specification as one of ordinary skill in the art would interpret these terms.

**Examiner's Response**

The terms are terms used by artisans of ordinary skill in the art and were interpreted in view of the Specification.

**Applicant's Argument for Modeling**

To the contrary, the Examiner in supposedly "interpreting" the term "modeling" only refers to sections of the Martin reference, discussed hereafter. The Examiner does not interpret the term "modeling" in light of the present specification. Thus, the claims are not bound by the Examiner's alleged "interpretation."

**Examiner's Response**

Art Unit: 2124

The terms are terms used by artisans of ordinary skill in the art and were interpreted in view of the Specification. Apparently the Applicant is unable to distinguish the term from the Specification. No specific part of the Specification has been identified as holding a different meaning and no lexicon has been provided.

**Applicant's Argument for Flow Control**

Furthermore, with regard to the "interpretation" of the term "Flow Control", the Examiner admits that Applicants do not claim flow control. Thus, the claims cannot be limited by the Examiner's "interpretation" of the term "Flow Control." With regard to whether what is recited in the claims may be "a product" of "flow control," this is immaterial to whether the claims as a whole are directed to patentable subject matter. Flow control is not claimed and thus, the claims should not be limited to the Examiner's interpretation of the meaning of flow control.

**Examiner's Response**

The Applicant seems confused why the term flow control was introduced. The Applicant did not explicitly claim "Flow Control". However, the claims do impact and have an impact on flow control. This is why the term was mentioned. It appears the Applicant is matching terms and not concepts. To make claim to an invention that contains rules in a programming environment and attempt to distance themselves from how the invention impacts flow control is not logical. The meaning of flow control and the activation of a control point are key issues in the following interaction.

**Applicant's Argument for Decorating**

Art Unit: 2124

With regard to the Examiner's "interpretation" of the term "decorating," the present specification on page 4 clearly describes a mechanism by which a decorator pattern is used to add a new behavior to an object to thereby generate a "decorated" object. Thus, the term "decorating" in claim 3 should be interpreted in light of the specification, not the personal interpretations of the Examiner.

**Examiner's Response**

The Examiner did interpret the term in view of the Specification. The Applicant has been provided another opportunity to make a distinction but apparently is unable to do so. Pointing to the Specification and describing the difference could distinguish the term.

**Applicant's Argument**

In summary, none of the Examiner's personal interpretation of the terms above may be used to limit the scope of the pending claims for the reasons noted above.

**Examiner's Response**

The Examiner has gone on the record and made the record clear the meaning of the terms as they are interpreted during the prosecution. The Applicant has attempted to distance themselves from the interpretations but has failed to provide any substance on why they differ from the Examiner's interpretation. Apparently the Applicant is unable to do so.

**II. Arguments Direct to the rejection under 35 U.S.C. § 102(b)**

**Applicant's Argument Directed Toward Claim 1**

Art Unit: 2124

The Office Action rejects claims 1-98 under 35 U.S.C. § 102 as being anticipated by Martin, "Principles of Object-Oriented Analysis and Design," published June 1, 1992. This rejection is moot with regard to the canceled claims and is respectfully traversed with regard to the remaining claims. With regard to claim 1, the Office Action states:

Martin anticipates a computer implemented process for applying a set of rules (Martin, Chapter 10, RULES, and page 138-139 and 249-251), the process comprising:

(a) placing a pre-method control before logic of a method (Martin, page 142, operation precondition) and post method control point after the logic of the method (Martin, page 142, post condition)

(b) associating a set rules with each control point (Martin, page 142, 147 "Operation" as per (a) above) based on a class of object in which the method resides (Martin, page 143, "...rules associated with diagrams of 00..."), name of the method and type of control point, whether the premethod control point or the post-method control point (Martin, page 142, operation precondition);

(c) invoking the method (Martin, page 116), wherein encountering each control point during the execution of the method comprises (Martin, page 142, post condition)

(i) determining if the encountered control point is active (Martin, page 122, IF structure in center diagram);



Art Unit: 2124

(ii) on the basis of an active control point (Interpreted as the result of the IF structure above further described in Appendix A on page 381

Control Conditions);

1) selecting rules based on a set of rules associated with the active control point associated in step (Martin, page 122, first diagram example is the control condition to fire missiles)(b);

2) running the selected rules (Martin, page 122, rule that lead to the control condition);

3) obtaining results from running the rules (Martin, page 122, trigger rule at the bottom of the page); and

4) combining the results using a combining algorithm specified by the control point (Martin, page 122, A control condition can function as a combining algorithm as seen in diagram in middle of the page and page 126 Figure 9.9 and Martin teaches a way to have a combining algorithm where one of three operations are selected as on page 124, and Martin teaches a way to have a combining algorithm where one can be selected as taught in the mutually exclusive notation on the bottom of the page 125).

### **Examiner's Response**

The Martin reference teaches programming constructs in an Object Oriented CASE tool environment. The examples are not limited to those examples in that exact format. To impose such a limitation on the Martin reference would require the Examiner to do the same to the invention which is also programming constructs. Martin teaches a mutually exclusive notation and non exclusive as per Appendix A.

Art Unit: 2124

**Applicant's Choice of Case Law**

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. In re Bond, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. In re Lowry, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. Kalman v. Kimberly-Clark Corp., 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). Applicants respectfully submit that Martin does not identically show each and every feature of the pending claims arranged as they are in the claims. Furthermore, Applicants respectfully submit that the claims do not read on the processes specifically described in the Martin reference and that the Office Action is engaged in applying broad teachings of Martin without regard for the actual teachings of Martin or the specific features recited in the claims.

**Examiner's Response**

The case law selected is Kalman v. Kimberly-Clark Corp., 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983) a case involving "filtering ribbon across a passage" not programming constructs which by their very intent are designed to be reused and in many different orders. To accept this case law as relevant would be to place severe limitations on the Applicant's invention. The Examiner does not find the use of this case law persuasive.

**Applicant's Argument**

Art Unit: 2124

Chapters 9 and 10 of Martin, which are the basis for the Office Action's rejection of all of claims 1-98, teach a method of modeling the behavior of an object oriented system. The modeling involves representing operations, preconditions of the operations, post conditions of the operations, control conditions, events and triggers. In the modeling described by Martin, an operation may have a precondition that identifies what must happen before the operation executes, and a post condition that describes the result of the operation if the operation executes with the precondition being satisfied. The operation may further include a control condition which is a condition that is used to determine whether the operation is to execute or not. The control condition must be checked prior to invoking the operation and may be a complex collection of Boolean conditions (page 122). Martin further teaches the implementation of these preconditions, post conditions and control conditions as rules in Chapter 10. Martin does not teach "associating a set of rules with each control point based on a class of object in which the method resides, name of the method and type of control point," "determining if the encountered control point is active," and "selecting rules based on a set of rules associated with the active control point associated in step (b)," as recited in claim 1.

#### **Examiner's Response**

The disagreement appears to be from the Applicant is claiming the determination of a control point as "active" and the Examiner has interpreted this to be a factor of flow control. The conditions that influence the path a program will take. This is inherent in programming and not able to be separated out. With this clarification the rejection might be better understood.

Art Unit: 2124

**Applicant's Argument**

With regard to the feature of "associating a set of rules with each control point based on a class of object in which the method resides, name of the method and type of control point," the Office Action alleges that this feature is taught by Martin on pages 142, 143, and 147. However, these sections of Martin have nothing to do with associating a set of rules with a control point based on a class of object in which the method resides, name of the method and type of control point. Page 142 merely describes that the event diagram is an executable diagram from which program code may be generated using a tool such as the OO-CASE tool. Page 143 merely describes that rules may be either object state rules or object behavior rules; that rules may be used with other types of diagrams other than event diagrams; and that rules may be stated as English expressions which may then be used to generate code. Page 147 merely shows examples of an operation, event diagram and state transition diagram that have rules attached to them.

**Examiner's Response**

The reference explicitly states the diagrams generate code not "might" as the Applicant has stated.

**Applicant's Argument**

There is nothing in these sections, or any other sections, of Martin that can remotely be considered to teach the feature of "associating a set of rules with each control point based on a class of object in which the method resides, name of the method and type of control point." It is not clear where or how the Office Action can extract any teaching from Martin that even suggests

Art Unit: 2124

the features of the presently claimed invention. Rather, it appears that the Office Action is engaged in taking broad teachings of event diagrams and generating code from event diagrams, and reading into these broad teachings the very specific features recited in Applicants' claims. However, the Office Action cannot read into the prior art features that are only present in Applicants' own disclosure and use the "modified" prior art to support a rejection of the claims. The Office Action must base its rejection solely on the teachings of the references themselves. In this case, there is nothing in the Martin reference that can be used to support a position of anticipation with regard to the above feature.

**Examiner's Response**

Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

**Applicant's Argument**

Similarly, there is nothing in the Martin reference that remotely even hints at the feature of "determining if the encountered control point is active." The Martin reference does not even recognize a possibility of having active or inactive control points. The Office Action equates the operation precondition and post conditions to the control points recited in the claims, even though they are not the same as the recited control points. A control point, as defined in the present specification is a point at which rules may be attached to add additional functionality. The

Art Unit: 2124

preconditions and post conditions of Martin are merely requirements for the operation to execute properly.

**Examiner's Response**

The Martin reference teaches a plurality of programming constructs. The Applicant seems to be limiting the use of the constructs. Limitations not stated in the reference.

**Applicant's Argument**

However, assuming that the preconditions and post conditions are the same as a control point, arguendo, Martin describes the precondition and post conditions as always having to be satisfied in order for proper operation of the object oriented system model. Thus, the precondition and post conditions are always "active" and there is no need to determine if they are active enough. It is for this reason that Martin does not mention anywhere in the entire reference, any step of determining if a control point is active.

**Examiner's Response**

The disagreement appears to be from the Applicant is claiming the determination of a control point as "active" and the Examiner has interpreted this to be a factor of flow control. The conditions that influence the path a program will take. This is inherent in programming and not able to be separated out. With this clarification the rejection might be better understood.

**Applicant's Argument**

The Office Action alleges that this feature is taught in Martin at page 122 simply because Martin teaches an IF structure. The Office Action equates the precondition with the control point

Art Unit: 2124

and then states that the presence of an IF structure in the precondition is the same as determining if a control point is active. This does not make any sense since, as is clearly described in the present invention and recited in the claims, if a control point is not active, the rules of the control point are not executed. If the precondition of Martin were the same as a control point, then the precondition must always be active in order for the IF structure to even operate. Therefore, the IF structure would always indicate that the precondition is active and thus, there is no need for the IF structure. Thus, the IF structure is not the same as the step of determining if a control point is active. Again, the Office Action is reading in teachings from the present specification into general descriptions provided by Martin with out any regard for what is actually being claimed and what Martin actually teaches.

#### **Examiner's Response**

The disagreement appears to be from the Applicant is claiming the determination of a control point as "active" and the Examiner has interpreted this to be a factor of flow control. The conditions that influence the path a program will take. This is inherent in programming and not able to be seperated out. With this clarification the rekjection might be better understood.

#### **Applicant's Argument**

Just as with the above, Martin also provides no teaching that is remotely similar to the feature of "selecting rules based on a set of rules associated with the active control point associated in step (b)." While Martin teaches that rules may be used to implement the precondition and post condition of an operation, there is no teaching in Martin of associating a

Art Unit: 2124

set; of rules with an active control point and then selecting rules based on this set of rules. Martin provides no teaching at all regarding selection of rules and the Office Action has not pointed out with particularity any section of Martin that teaches a selection of rules based on a set of rules associated with the active control point.

**Examiner's Response**

The disagreement appears to be from the Applicant is claiming the determination of a control point as "active" and the Examiner has interpreted this to be a factor of flow control. The conditions that influence the path a program will take. This is inherent in programming and not able to be separated out. With this clarification the rejection might be better understood.

**Applicant's Argument**

The Office Action alleges that this feature is taught by Martin in the first diagram on page 122 (although the following text indicates that the Examiner is in actuality referring to the last figure on page 121). These diagrams merely illustrate the use of control conditions that are checked prior to an operation executing. There is no selection of rules even mentioned or shown in these figures, let alone the selection of rules based on a set of rules associated with an active control point. The Office Action is referencing portions of the Martin reference that do not even have anything to do with the features of the claim.

**Examiner's Response**

The Applicant seems to be arguing that flow control would not be a factor as to which rule would be selected.



Art Unit: 2124

**Applicant's Argument**

It is not understood how the figures referred to can be interpreted in any way to teach the selection of rules based on a set of rules associated with an active control point. The figure illustrating firing of a missile is provided to show how a control point may be used to handle a plurality of triggers that are required for the operation to execute. The first diagram on page 122 also illustrates this concept. However, nowhere in the figures of the accompanying text is there anything mentioned about selection of rules based on a set of rules associated with an active control point.

**Examiner's Response**

The disagreement appears to be from the Applicant is claiming the determination of a control point as "active" and the Examiner has interpreted this to be a factor of flow control. The conditions that influence the path a program will take. This is inherent in programming and not able to be separated out. With this clarification the rejection might be better understood.

**Applicant's Argument**

On page 122 Martin does state that the control condition may be a collection of "or" conditions applied to triggers (see the middle diagram on page 122). However, the "or" Boolean condition does not constitute a selection of rules based on a set of rules associated with an active control point. Rather, the "or" conditions merely operate to state that if any one of the triggers satisfies the control condition, then the operation will execute. There is no selection of rules.

Art Unit: 2124

Thus, Martin does not teach each and every feature of claim 1 as is required under 35 U.S.C. § 102(b). Accordingly, Applicants respectfully request withdrawal of the rejection of claim 1 under 35 U.S.C. § 102(b).

**Examiner's Response**

The disagreement appears to be from the Applicant is claiming the determination of a control point as "active" and the Examiner has interpreted this to be a factor of flow control. The conditions that influence the path a program will take. This is inherent in programming and not able to be separated out. With this clarification the rejection might be better understood.

**Applicant's Argument directed toward Claim 2**

With regard to claim 2, Martin does not teach "defining a control point just before logic of at least one method, wherein the control point is a method context control point." A method context control point is a control point that may have a variety of different rules and different types of rules associated with it as the need for these rules changes over time (see page 11, lines 17-29 of the present specification). Martin does not teach such method context control points. Rather, Martin teaches preconditions, post conditions, and control conditions that perform specific functions that are not intended to change over time and are not intended to have different rules and different types of rules associated with them. This is because these conditions are linked to the operation so closely that the conditions cannot be changed without causing the operation to not execute properly.

**Examiner's Response**

Art Unit: 2124

Persuasive.

**Applicant's Argument**

The method context control point of the present invention allows the object model to be modified without having to change the underlying model itself. By applying new rules at active control points, new functionality is available without having to modify the model. With the event diagrams of Martin, the model itself must be modified, i.e. the operations and their preconditions, post conditions and control conditions, in order to obtain new functionality. Thus, the present claims do not read on the Martin reference and the Martin reference does not anticipate the presently claimed invention.

**Examiner's Response**

Claim 2 was allowed for reasons other than this argument

**Applicant's Argument**

Since Martin does not teach method context control points, Martin does not teach each and every feature recited in claim 2 as is required under 35 U.S.C. § 102(b). At least by virtue of their dependency on claim 2, Martin also does not teach or suggest the features recited in dependent claims 3-11. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 2-11 under 35 U.S.C. § 102(b).

**Examiner's Response**

Persuasive

**Applicant's Argument**

Art Unit: 2124

Martin also does not teach the step of "externalizing the at least one control point as a class and instantiating it at the at least one control point," as recited in claim 6. The Office Action equates the precondition and post condition of an operation as the same as a control point. However, the precondition and post condition of an operation are not a class and are not instantiated. The precondition and post condition of Martin are tied to an operation, which may be represented as a method. Thus, the precondition and post condition cannot be classes and cannot be instantiated.

The Office Action alleges that this feature is taught at pages 133-136 and page 167. However, there is nothing in these, or any other sections, of the Martin reference that remotely comes close to a control point that is externalized as a class which is instantiated at a control point. The Office Action is again engaged in reading into Martin teachings that simply are not there and are only present in Applicants' disclosure.

**Examiner's Response (emphasis)**

The Examiner believes this to be the Applicant's best argument but the argument merely repeats claims language and does not point to the Specification and describe the difference they are attempting to make. Examiner encourages an After Final Argument in a format that points to the Specification and clearly describes the differences. This could lead to additional Allowable subject matter.

**Applicant's Argument**

Art Unit: 2124

Martin also does not teach "defining, with a control point, at least one of a rule selecting algorithm and a rule results combination algorithm," as recited in claim 10. The Office Action does not even state where this teaching may be found in Martin but merely refers to the rejection of claim 1. There is nothing in Martin that provides for a selection algorithm that is defined for a control point. This is because rules cannot be selected for a precondition, post condition or control condition. Rather, the rules are assigned to these conditions and are fixed thereafter. There is no selection algorithm for selecting amongst rules that are associated with a condition in Martin.

#### **Examiner's Response**

The disagreement appears to be from the Applicant is claiming the determination of a control point as "active" and the Examiner has interpreted this to be a factor of flow control. The conditions that influence the path a program will take. This is inherent in programming and not able to be separated out. With this clarification the rejection might be better understood.

#### **Applicant's Argument**

With regard to claim 11, Martin does not provide for changing of rules associated with a control point in a set of rules. The Office Action alleges that this feature is taught by Martin simply because it teaches the use of OO-CASE and I-CASE. However, when rules are assigned to a precondition, post condition or control condition, as taught by Martin, they are fixed and are not changeable without destroying the proper functioning of the associated operation. This is because the precondition must be satisfied in order for the operation to execute and the post condition

Art Unit: 2124

must be satisfied in order for the operation to execute properly. Thus, if these are changed, the operation will execute improperly.

**Examiner's Response**

The disagreement appears to be from the Applicant is claiming the determination of a control point as "active" and the Examiner has interpreted this to be a factor of flow control. The conditions that influence the path a program will take. This is inherent in programming and not able to be separated out. With this clarification the rejection might be better understood.

**Applicant's Argument**

Claims 12, 46, 48, 51, 70, 72, 75, 94, 96 define over Martin for similar reasons as noted above with regard to claim 2. In particular, Martin does not teach a method context control point as recited in each of these claims and as discussed above. At least by virtue of their dependency on claims 12, 46, 48, 51, 70, 72, 75, 94 and 96, Martin does not teach the features recited in dependent claims 13-15, 49-50, 52-59, 73-74, 76, 79-83, 97-98. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 12-15, 46, 48-59, 72-74, 75-76, 79-83, 94 and 96-98 under 35 U.S.C. § 102(b).

**Examiner's Response**

Some of these claims do not appear to contain the limitation found allowable in claim 2. Claims 2, 48, 70 and 94 are allowable.

**Applicant's Argument**

Art Unit: 2124

In addition to the above, Martin also does not teach many of the features recited in the dependent claims set forth above. For example, Martin does not teach that the rules perform at least one function that varies over time, as recited in claim 14. As mentioned above, the precondition, post condition and control condition are only used to determine whether an operation is to execute and whether an operation has executed correctly. There is nothing in Martin that teaches a control point that has rules that perform a function that varies over time. None of the functions of the precondition, post condition or control condition in Martin vary over time.

**Examiner's Response**

The clock event programming construct in Martin is not limited.

**Applicant's Argument**

The Office Action alleges that this feature is taught by Martin at page 117 simply because Martin teaches clock events and that Martin teaches associating rules with event diagrams. The clock events have nothing to do with preconditions, post conditions and control conditions. The clock events are separate entities in the event diagram from preconditions, post conditions and control conditions. The Office Action is again taking general teachings and conjuring up specific teachings regardless of the actual teachings of Martin. The only place that control points having rules that perform functions that vary over time are even mentioned is in Applicants' own disclosure. While clock events are triggered based on time, this in no way teaches a control point that has a rule that performs a function which varies over time.

Art Unit: 2124

**Examiner's Response**

The clock event programming construct in Martin is not limited. The persuasive concept of "vary over time" was persuasive when claimed like in claim 94.

**Applicant's Argument**

With regard to claims 49, 73 and 97, Martin does not teach controlling a method by exiting the method. As noted above, the operation in Martin may be implemented as a method. For the method to be invoked, the precondition must be satisfied and the control condition must be satisfied. Thus, the only options are that the operation is either executed because both the precondition and the control conditions are satisfied or the operation is not executed because one or both of the precondition and control condition are not satisfied. There is no possibility of "exiting" the operation based on the precondition and the control condition because the operation is not entered until after the precondition and control condition operate. Thus, Martin does not teach controlling the method by exiting the method as described in claims 49, 73 and 97.

**Examiner's Response**

Exiting operations are inherent in programming.

**Applicant's Argument**

Regarding claims 55 and 79, Martin does not teach flagging means for flagging the at least one control point on the basis of being active. As previously mentioned, Martin does not even



Art Unit: 2124

recognize that a control point may be active or non active. Thus, Martin cannot flag a control point based on whether it is active.

**Examiner's Response**

See the argument related to "active" control point above.

**Applicant's Argument**

The Office Action merely points to its rejection of claim 1 in rejecting this feature. However, none of the sections cited in the rejection of claim 1, nor the rejection itself, address anything remotely resembling flagging a control point on the basis of the control point being active. Martin does not even discuss active and non active control points.

**Examiner's Response**

See the argument related to "active" control point above.

**Applicant's Argument**

With regard to claims 56, 58, 80 and 82, Martin does not teach defining a rule selection algorithm associated with the at least one control point. As previously mentioned above, Martin does not teach any mechanism for selecting rules, let alone a rule selecting algorithm that is associated with a control point. The Office Action alleges that Martin teaches this feature at page 168. However, there is nothing on page 168, or in any other section of Martin, that teaches or even remotely suggests a rule selection algorithm that is associated with a control point. The Martin reference merely teaches the association of rules with event diagram preconditions, post

Art Unit: 2124

conditions and control conditions. There is no selection mechanism associated with these preconditions, post conditions and control conditions.

**Examiner's Response**

See the argument related to "active" control point above.

**Applicant's Argument**

With regard to independent claim 16, Martin does not teach defining rules to the at least one control point on the basis of the object's class name, method name, and position of the; at least one control point in the method. The lack of teachings in Martin with regard to this feature are discussed in detail in the rejection of claim 1 above and the claim 16 defines over Martin for these same reasons.

**Examiner's Response**

Same reasons as per claim 1.

**Applicant's Argument**

Thus, Applicants respectfully submit that Martin does not teach each and every feature recited in claim 16 as is required under 35 U.S.C. § 102(b). At least by virtue of their dependency on claim 16, Martin also does not teach the features recited in dependent claims 17-22. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 16-22 under 35 U.S.C. § 102(b).

**Examiner's Response**

Same arguments as per above.

Art Unit: 2124

**Applicant's Argument**

In addition to the above, Martin also does not teach many of the specific features set forth in dependent claims 17-22. For example, Martin does not teach activating at least one control point having associated rules, as recited in claim 17. As discussed in depth above, Martin does not even recognize the possibility of having active and nonactive control points, let alone activating a control point.

**Examiner's Response**

See the argument related to "active" control point above.

**Applicant's Argument**

Similarly, this lack of teaching in Martin applies to claim 22 which recites "deactivating the at least one control point." Again Martin does not teach active or nonactive control points and thus, cannot teach deactivating a control point.

**Examiner's Response**

See the argument related to "active" control point above.

**Applicant's Argument**

With regard to claims 23 and 27, Martin does not teach a method context control point, as previously discussed in detail above. The rejection of claims 23 and 27 is overcome based on the same reasoning as set forth above with regard to claims 2, 12, 46, 48, 51, 70, 72, 75, 94 and 96.

**Examiner's Response**

Some of these claims were found allowable but not for the limitation argued.

Art Unit: 2124

**Applicant's Argument**

Thus, Applicants respectfully submit that Martin does not teach each and every feature recited in claims 23 and 27 as is required under 35 U.S.C. § 102(b). At least by virtue of their dependency on claims 23 and 27, respectively, Martin also does not teach the features recited in dependent claims 24-26, 28 and 31-35. Accordingly, Applicants respectfully request withdrawal of the rejections of claims 23-28 and 31-35 under 35 U.S.C. § 102(b).

**Examiner's Response**

Apparently already argued and answered above. No new argument present.

**Applicant's Argument**

In addition to the above, Martin does not teach many of the features recited in the dependent claims 24-26, 28 and 31-35. For example, Martin does not teach flagging the at least one control point on the basis of being active, as recited in claim 31. This feature was discussed in detail above with regard to claims 55 and 79.

**Examiner's Response**

See the argument related to "active" control point above.

**Applicant's Argument**

Martin also does not teach a rule selection algorithm associated with at least one control point as recited in claims 32 and 34. This feature was discussed in detail above with regard to claims 56, 58, 80 and 82.

**Examiner's Response**

Art Unit: 2124

See the argument related to “active” control point and flow control above.

**Applicant’s Argument**

With regard to claims 36, 60 and 84, Martin does not teach associating a rule with another method within an object class. As previously mentioned above, Martin only teaches associating rules with preconditions, post conditions and control conditions in an event diagram. There is no teaching in Martin that the same rule may be associated with two different methods in an object class. While similar rules may be associated with different conditions in an event diagram, there is nothing in Martin that teaches that the same rule may be associated with two methods within an object class.

**Examiner’s Response**

The ability to call “helper” methods within an object is considered inherent and present in languages such as ANSI standard C++.

**Applicant’s Argument**

The Office Action alleges that this feature is taught on pages 266-268. The Office Action alleges that this section of Martin teaches associating rules with more than one object, reuse and inheritance which can all be read on the claimed feature. However, the general teachings of reuse and inheritance do not in themselves teach the specific features recited in claims 36, 60 and 84. There is nothing in Martin that teaches or even suggests to one of ordinary skill in the art that the same rule can be associated with a plurality of methods, as recited in claims 36, 60 and 84. The

Art Unit: 2124

Office Action is again engaged in reading into general teachings, the specific teachings found only in Applicants' disclosure without any basis for such reading in of teachings in the reference itself.

**Examiner's Response**

Reuse is one of the benefits of Object Oriented programming. The idea one would not associate rules with one or more object is not consistent with object oriented languages and the teachings of Martin.

**Applicant's Argument**

Thus, Applicants respectfully submit that Martin does not teach each and every feature recited in claims 36, 60 and 84 as is required under 35 U.S.C. § 102(b). At least by virtue of their dependency on claims 36, 60 and 84, Martin also does not teach the features recited in dependent claims 37-39, 41, 61-63, 65, 85-87 and 89. Accordingly, Applicants respectfully request withdrawal of the rejection of claim 36-39, 41, 60-63, 65, 84-87 and 89 under 35 U.S.C. § 102(b).

**Examiner's Response**

As argued prior.

**Applicant's Argument**

Regarding claim 42, Martin does not teach the feature of determining if a control point is active. This feature has been address above with regard to claim 1 and thus, claim 42 defines over Martin for similar reasons as noted above with regard to this feature in claim 1.

Art Unit: 2124

Thus, Applicants respectfully submit that Martin does not teach each and every feature recited in claim 42 as is required under 35 U.S.C. § 102(b). At least by virtue of their dependency on claim 42, Martin does not teach the features recited in claims 43-44. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 42-44 under 35 U.S.C. § 102(b).

In addition to the above, Martin also does not teach the specific features recited in claims 43-44. For example, Martin does not teach a selecting algorithm associated with an active control point, as discussed at length above.

This same distinction over Martin also applies to independent claim 45 and thus, claim 45 is allowable over the Martin reference for this same reason. With regard to claim 46, Martin does not teach a method context control point, as discussed above. Regarding claim 47, Martin does not teach determining if a control point is active. Thus, Applicants respectfully request withdrawal of the rejections of claims 45-47 under 35 U.S.C. § 102(b).

With regard to claim 66, Martin does not teach determining if a control point is active, as previously discussed above with regard to claim 42 and others. Thus, Martin does not teach each and every feature recited in claim 66 as is required under 35 U.S.C. § 102(b). At least by virtue of their dependency on claim 66, Martin also does not teach the features recited in dependent claims 67 and 68. In addition, Martin does not teach a selecting algorithm as recited in claim 67. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 66-68 under 35 U.S.C. § 102(b).

Art Unit: 2124

With regard to claim 69, again Martin does not teach a selecting algorithm associated with a control point. Regarding claim 70, Martin does not teach a method context control. point. With regard to claim 71, Martin does not teach determining if control points are active. Thus, Martin does not teach each and every feature of claims 69-71. Accordingly, Applicants respectfully request withdrawal of the rejections of claims 69-71 under 35 U.S.C. § 102(b).

Regarding claims 90, 92 and 95, Martin does not teach determining if a control point is active. With regard to claims 91 and 93, Martin does not teach a selecting algorithm. Accordingly, Applicants respectfully request withdrawal of the rejections of claims 90-93 and 95 under 35 U.S.C. § 102(b).

#### **Examiner's Response**

It appears we have reached a level which dependent claims are relying on prior non persuasive arguments.

#### ***Correspondence Information***

9. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to **Todd Ingberg** whose telephone number is **(703) 305-9775**. The Examiner can normally be reached on Monday through Thursday from 6:30 a.m. to 5:00 p.m. If attempts to reach the examiner by telephone are unsuccessful, the **Examiner's Supervisor, Gregory Morse** be reached at **(703)308-4789**. Any response to this office action should be mailed to: **Director of Patents and Trademarks Washington, D.C. 20231**, or **Hand-delivered** responses should be brought to **Crystal Park II, 2121 Crystal Drive Arlington, Virginia,**



Art Unit: 2124

**(Receptionist located on the fourth floor), or faxed. The following fax numbers apply:**


**After Final (703) 746 - 7238**

**Official (703) 746 - 7239**

**Non Official/ Draft (703) 746 - 7240**

**Todd Ingberg**

May 5, 2002

  
**GREGORY MORSE**  
**SUPERVISORY PATENT EXAMINER**  
**TECHNOLOGY CENTER 2100**